

JOSS: 20,000 hours at a console—a statistical summary

by G. E. BRYAN
The RAND Corporation
Santa Monica, California

INTRODUCTION

JOSS* is a special-purpose computing system designed to provide users with a substantial and highly interactive computational capability.¹ The first JOSS system, developed for the JOHNNIAC computer machine by J. C. Shaw, was operational in early 1963. Work on an expanded system utilizing a modern PDP-6 computer began in 1964, and the system became operational in February 1966. Although there are many systems today that provide time-shared access to a computer, little is known of precisely how such machines are used. This was especially apparent at the beginning of JOSS development. Substantial effort was therefore made to provide a measuring or instrumenting capability within the system not only to record use of the system as a whole but also to record characteristics of use for individual users of the system. This paper presents the first results of these metering efforts.

To properly interpret these results, it is necessary to be familiar with the operation of the system, the features provided to users, and the hardware on which the system is implemented.

Figure 1 shows the essential hardware of the PDP-6 computer on which JOSS is implemented. The central processor is of modern design, typically executing instructions in $5\mu\text{sec}$. A seven-level priority interrupt system is used for the control of input/output actions. Two independent 16,000-word core memories operate at $1.75\mu\text{sec}$ cycle time. A one-million-word drum is used for secondary storage of user programs. It has independent access to user core memory, can transfer 4000 words to memory in 17 milliseconds ($4\mu\text{sec}$ each word), and rotates in 35 milliseconds. The organization is such that 60 milliseconds elapsed time is re-

quired to perform a complete user interchange between drum and core.

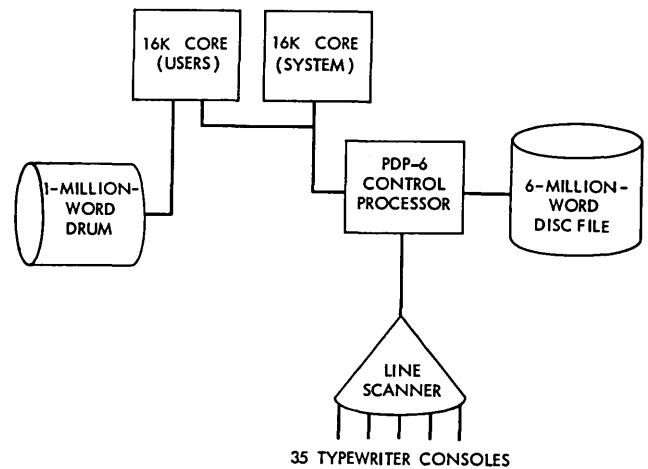


Figure 1—PDP-6 central processor

The transfer of information between drum and core is accomplished without interference in the use of core by drum and central processor references. No interference occurs because all software instructions are contained in a separate memory and thus have a maximum reference rate of once per instruction (about $4\text{--}5\mu\text{sec}$) to the user memory. The maximum reference rates of drum and central processor to the user memory box, therefore, are easily accommodated by its $1.75\mu\text{sec}$ cycle time.

The disc file is a tertiary store for users' programs and data. Although it is possible for users to chain their programs by means of the disc, it is used almost exclusively for long-term storage, that is, at a low rate compared to its capability.

A line scanner that contains a one-character buffer in each direction, for each user's typewriter, connects each console to the computer. Interrupts from the scanner signal the complete transmission or reception of

*JOSS is the trademark and service mark of The RAND Corporation for its computer program and services using that program.

each character. Currently, thirty-one specially designed typewriter consoles and six teletype consoles are connected to the system, although several times this number could be accommodated. Nine of the consoles are in remote locations operating over either private or dataphone lines, while the remainder are local to the computer with hard wire connections.

The system software, which consists of an interpreter for the JOSS language (the single offering of the system), is permanently resident in the system memory, together with software for I/O control and system supervision. Input/output for the users' consoles is buffered line by line through core areas in the system memory so that the user data need not be present in core during I/O delays caused by the consoles.

It is significant to note that no high-speed I/O devices such as tapes are available to users. Although the disc may be used for high-speed I/O, all data on the disc must have originated at typewriter consoles or have been generated by user programs.

System characteristics

Total usage of the JOSS system has increased rapidly since the PDP-6 version replaced the JOHNNIAC for regular JOSS service in February 1966, although the system has had its troubled times. Trends of total compute (central processor) time for users and the number of different users by month for the first year of PDP-6 operation are plotted in Figure 2 and point up

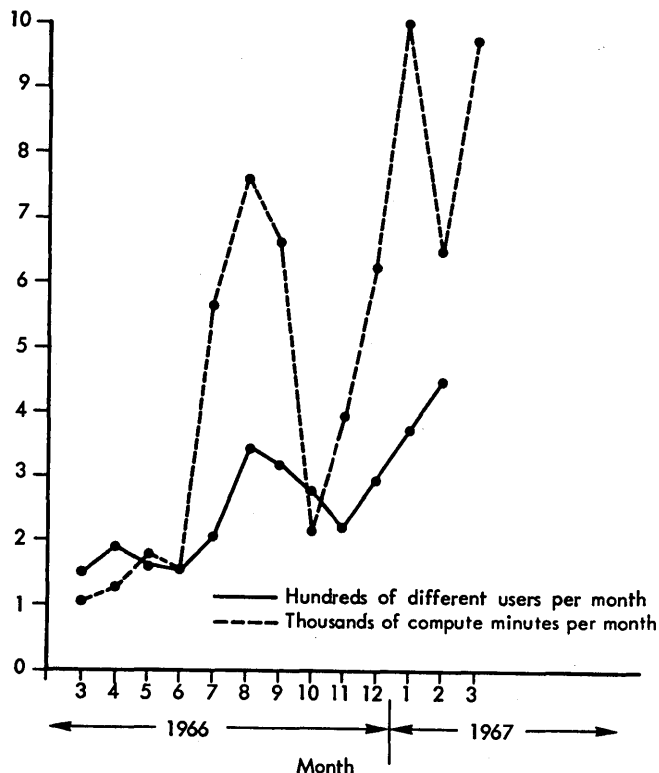


Figure 2—JOSS usage trends

several specific events. First, the activation of the disc and drum systems in July 1966 allowed a fivefold increase in the size of user programs and long-term storage of programs on the disc. Also, in the July-September period the number of consoles was almost doubled, from 18 to 30. Second, there were two sieges of machine trouble: The most serious lasted from late September through early November and a less serious one occurred in early February 1967.

It is clear that in a multiuser environment reliability is of critical importance. Every system failure directly inconveniences each user, and often, particularly in the case of remote consoles, users have very little idea of what has happened or when recovery is to be expected. This is in contrast with the batch mode of computer operation, where there is always the possibility of re-trying the job and continuing without the users' knowledge. To provide information to users during times of breakdown, as well as to provide scheduling and other general information, the JOSS system includes as an adjunct a recorded telephone answering service.

Speed of operation is of particular interest in JOSS because it is an interpretive system. Figure 3 presents a distribution of the average rate of statement execution as measured each minute. The mean rate is about 10 milliseconds per statement, with the average statement including about five arithmetic or function operations. Both statement scanning and the arithmetic itself are interpretive, the operation time being divided approximately evenly between these two functions. The estimates from instruction counts within the interpreter, as well as measurements on specific problems, indicate that the interpretation process is slower than

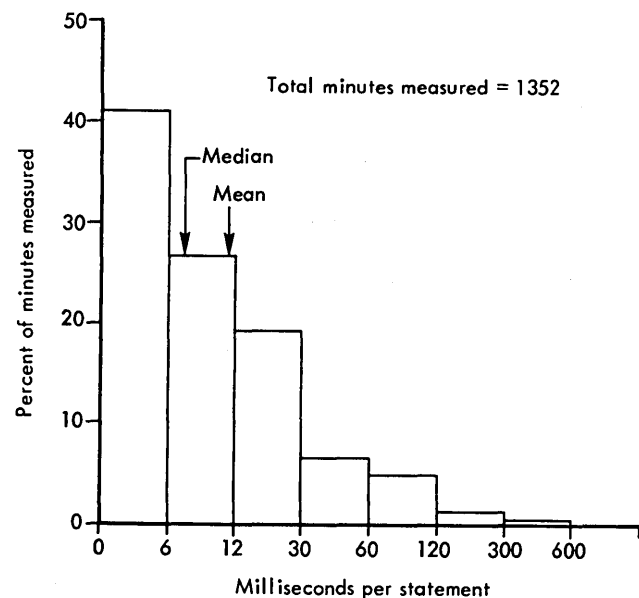


Figure 3—Statement interpretation rate

the equivalent compiled program by a factor of between 20 and 50.

Interpretation of a "typical" JOSS statement breaks down approximately as follows:

Interprogram step sequencing	1.0 ms
Execution of the action specified by the verb	1.75 ms
Five arithmetic or function operations (1.2 ms each)	6.0 ms
Interpretation time for typical statement	<u>8.75 ms</u>

In JOSS, statements may take far longer to execute than the times given above: for example, if they contain summations or nested functions of considerable complexity. However, compilation time which in many batch shops accounts for 25 percent of machine time, is eliminated. The elementary functions (e.g., *sin*, *log*, *exp*, etc.) and I/O conversion run at speeds comparable to compiled code.

Program storage in JOSS is half that required for compiled-out code, while data storage requires twice the space. Individual programs can vary these ratios by large amounts. JOSS code is carried in character form. For example, the string $a = b + c$ occupies one word, while a compiled-out version would require three words in most computers.

Data, on the other hand, require two words—including the floating decimal representation and a link to associated numbers. Because JOSS stores only array elements that have a *defined value* rather than assigning block storage to arrays and because the system has the ability to operate meaningfully on sparse arrays, certain programs may require less total storage for data in JOSS than an equivalent FORTRAN program.

Average character I/O rates are shown in Figures 4, 5, and 6. The rates are typical for an aggregate of users during a prime shift. The overall ratio of output to input on a line basis is 2.85.

The maximum I/O rate is 15 characters per second per user, which is determined by the typewriter characteristics. At any given time, an individual typewriter can only be used in one direction (though the console does signal for certain synchronizing purposes in both directions simultaneously). Thus, in engineering the real-time console interrupt facilities, the maximum *total* rate per console is 15 characters per second per user. The real-time problem is actually less severe, because delays on character output can only result in slower typing and not in lost characters. Input characters can be lost, but for the average rates shown the probability is exceedingly small until several hundred consoles are attached.

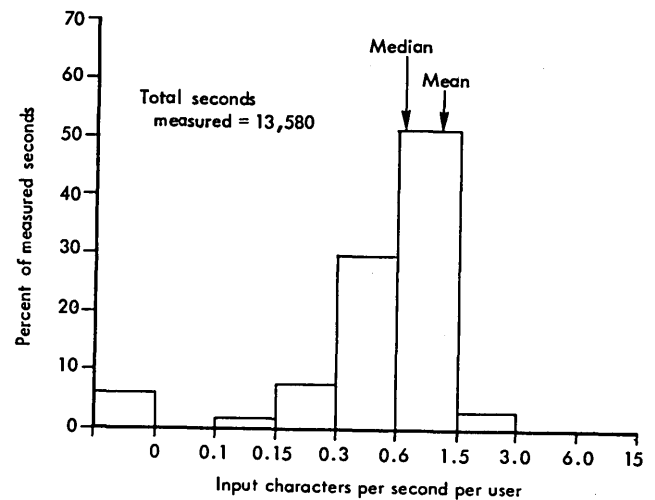


Figure 4—System input rate

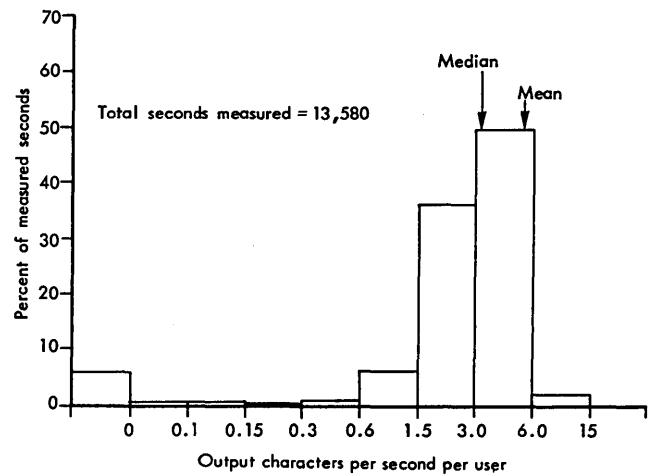


Figure 5—System output rate

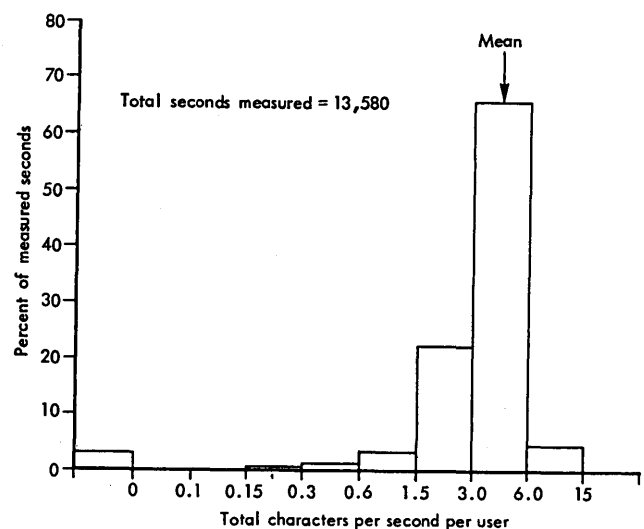


Figure 6—System input/output rate

The amount of core used, weighted by session time, is shown in Figure 7. The mean size is near 2000 words, and the shape of the distribution is similar to that measured in scientific FORTRAN batch-shop environments, although the median size in such environments is higher, namely, 4000 words.²

Program size is only a good measure of system load when weighted by execution time—large programs with only minor execution time do not provide substantial system load. In JOSS, when program size is weighted by execution time, the mean shifts upward from 2000 to 3000 words. The phenomenon of large programs tending to have longer running times has also been observed in batch FORTRAN environments. For example, an upward median shift from 4000 to 10,500 words was observed in one 1964 study² when the distribution of program size was weighted by program execution time. It is interesting to note that this same study observed *no* shift if only program instructions were considered. Thus, a more precise statement is that programs with a large amount of *data* tend to run a long time.

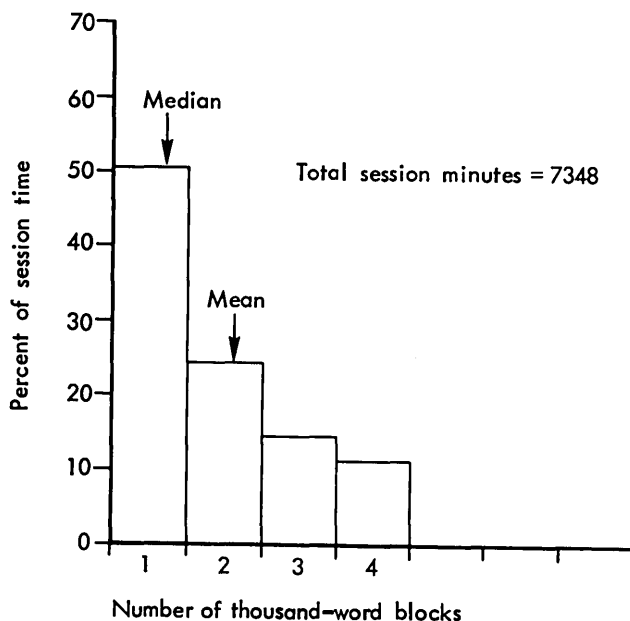


Figure 7—User program size

Overhead

Loss of computation power due to overhead functions is of concern on any computer and has been particularly critical in time-sharing systems. Modern equipment has eased the problem considerably by allowing simultaneous computing while performing swaps between high-speed memory and secondary storage. In JOSS, the limitation on user program size guarantees space for at least four user programs concurrently in

core. Since variable-size blocks of from one- to four-thousand-word units are assigned to users, the maximum in core is sixteen. From eight to ten is a typical figure. JOSS suffers essentially zero loss of compute power due to memory interference from, or waiting for, swaps. JOSS supervisory overhead for user scheduling, resource allocation, performance metering, and accounting functions requires $\frac{1}{4}$ to $\frac{1}{2}$ percent of real time. Moving users' data and programs within core memory to provide contiguous blocks of space for other users' programs being transferred from the drum, and checksum calculations for these transfers, are the major overhead items in the JOSS system. This is the price paid for variable-size user blocks in a machine without a paged memory. During periods of typical activity, overhead due to these functions uses about 5 percent of time, although brief periods (1 to 10 minutes) of very heavy activity may push the figure to between 30 and 40 percent.

Disc file usage

JOSS users maintain programs and data in the disc file on a long-term basis. Each user who so desires is assigned a file of up to 25 items, consisting, in aggregate, of at most 100 records of 128 words each. A user may have as many files assigned to him as he wishes. In March 1967, there were 330 files assigned to approximately 230 individuals and groups. Some large-user groups, particularly those outside RAND, operate with a pool of files assigned to them collectively: for example, the 100 students at the Air Academy.

In a typical session, users access the files about 5 times thrice to recall items, once to discard an item, and once to save an item. Thus, the total of 1000 actions each day, with perhaps 1000 words being transmitted in each action, gives an average rate of 40 words per second—a very modest load on the files.

The files have been in use since July 1966 and in March 1967 contained 9000 user records, which represents about 20 percent of the file capacity. In the first three months of 1967 the growth rate was steady at 1500 records per month, which if it continues, will fill the file in two years. Three factors may be responsible for this remarkably slow growth rate, which is at variance with experience in other time-shared installations. First, only one form of information is stored in the files—symbolic programs and data. In general-purpose systems, it is common for users to store absolute, relocatable, symbolic, and listing forms of the same program. Second, the limited size of the files and programs, together with the requirement to fill out a form (noncomputer) to get an additional file, tends to cause users to keep their files free of obsolete or no longer needed

items. Third, all input must come through users typewriters.

Characteristics of usage

Because JOSS is an interpretive system (in the machine a single program executes machine instructions that simulate the execution of JOSS language "programs"), there is a unique opportunity to measure the characteristics of both the JOSS users and the programs they generate. We can take advantage of this fact to record in detail many events—counting and timing statements, arithmetic operations, and user interactions. Even in the most expensive cases, counting actions require less than 1 percent of time.

A user session on JOSS spans the time from console turn-on to console turn-off. It corresponds to the log-on/log-off interval in most remote terminal systems. Figure 8 displays the distribution of session times for 227 sessions that occurred during one 24-hour period. The general form of the distribution remains quite constant over time with a mean in the range 45 to 50 minutes and median at 20 minutes. The length of the session may typify either the average computational needs of engineers and scientists or the amount of time it takes to solve problems that they are now willing to commit to JOSS.

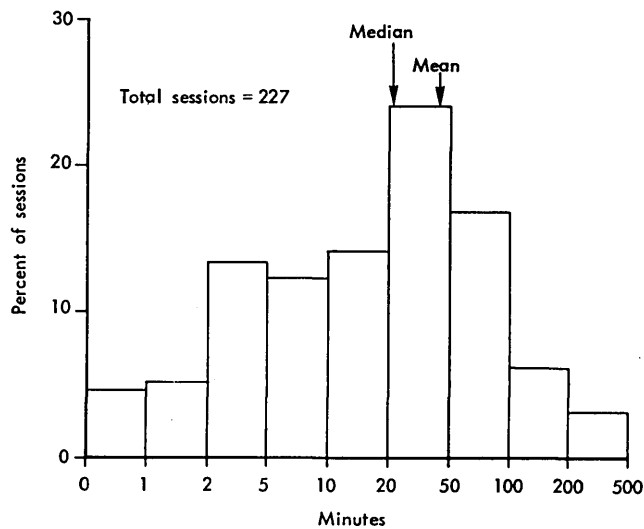


Figure 8—User session time

Program size, shown in Figure 9, is recorded at log-off time. Typical of the distribution is the peak at very small program sizes—10 cells will store only two or three program steps. The usage in this range represents the "desk calculator" usage of JOSS.

The mean amount of compute time used during each session is just under 4 minutes—in the same range as average compute times in scientific FORTRAN

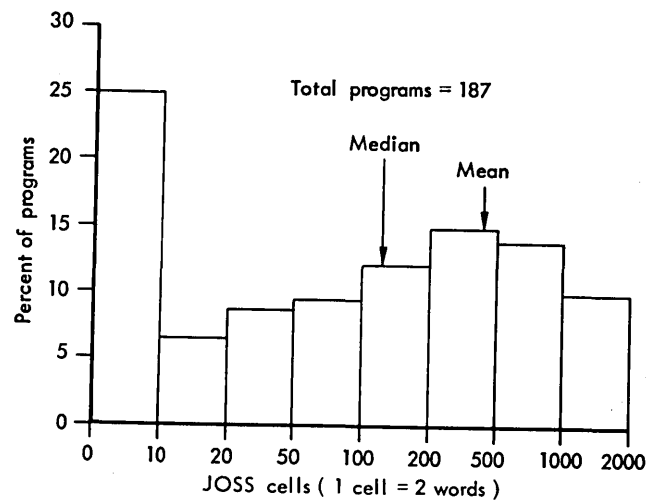


Figure 9—Used "size"

batch-processor shops. Figure 10 shows a typical distribution of compute time per user session. As in many computing environments, the mean compute time is misleading. For this sample, the mean is 6.6 minutes, while the median is 7 seconds, and 85 percent of all sessions use less than 50 seconds of computing. The general form of the distribution is typical: a high peak near the origin decaying rapidly at first and then very slowly leaving a long tail that contributes heavily to the average.

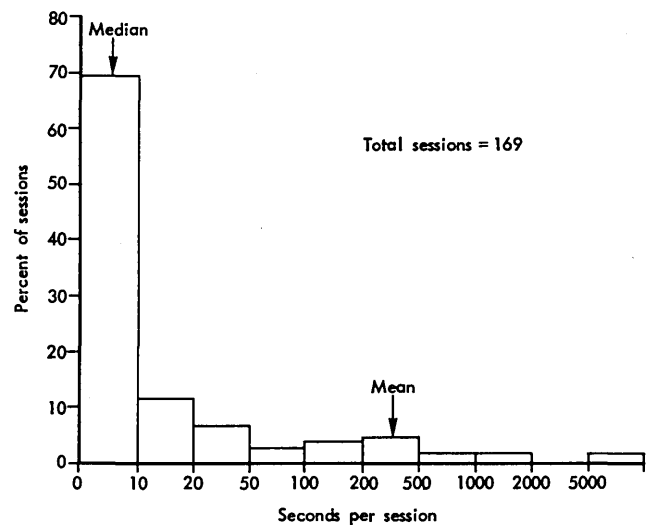


Figure 10—User compute time

The ability of a time-sharing system to respond rapidly to user requests depends heavily on the shape of this load curve. In the general-purpose, time-sharing systems,^{3,4} typical load curves have far fewer jobs with small compute requirements and far more jobs with large compute requirements. This results in long queues

for compute resources and, in turn, in long delays in response. On the other hand, typical JOSS loads are such that the length of the compute queue seldom exceeds a length of three or four even during the heaviest usage hours.

For purposes of measurement, we divide a user session into subtasks, each initiated by the release of a typed command to the system. The requested tasks can be quite simple, such as *Type 2 + 2.*, or more complex, such as those that initiate a computation of several hours' duration. During a typical session, the user hits carrier return 82 units, creating 82 tasks each involving, on the average,

1. Execution of 112 JOSS statements,
2. 500 arithmetic or function computations,
3. 2 seconds of compute time, and
4. 3 output lines.

As usual the averages are deceiving, as can be seen in Figure 11, which shows the distribution of compute time per task. The shape of the distribution is typical with a high initial peak and a very long tail. While the mean is 2 seconds, the median is 1/100th of that at 22 milliseconds.

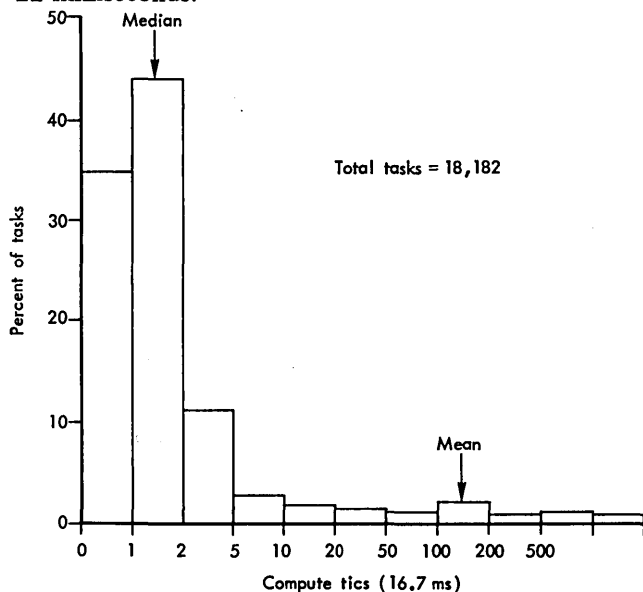


Figure 11—Compute time per task

The scheduling algorithm used in JOSS⁵ gives highest priority to carrier-return interrupts, that is, to the processing of commands directed from the user to the system. Up to one time quanta (200 milliseconds) is allowed for the command processing before the task is considered compute-bound and relegated to round-robin processing in the compute queue. As can be seen from the compute request distribution in Figure 11, well over 90 percent of the requests are completed on a high-priority basis within the first quanta.

Task turnaround time is measured from the time a task is initiated until the last of any requested output is produced at the console—in JOSS the time from carrier return until console control is returned to the user. Figure 12 presents a typical distribution of task

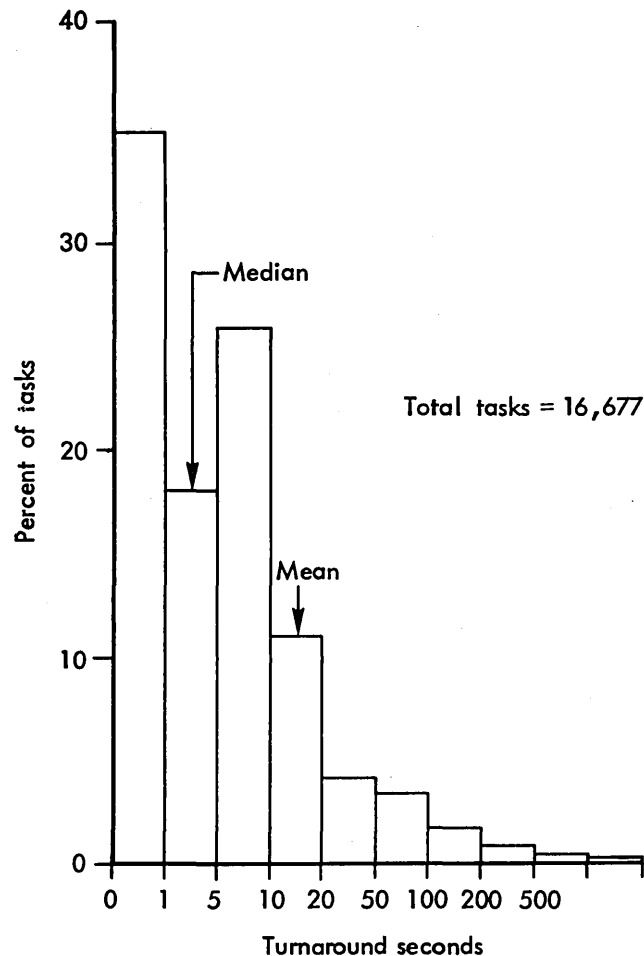


Figure 12—Task turnaround time

turnaround. Again, we have a mixture of a few rather long tasks with many short tasks. The mean turnaround time is 10 seconds including 7 seconds of typing and 2 seconds of computing. Some of the compute time is overlapped with the typing and some delayed because the computer is shared with other users. The *median* turnaround is 1.7 seconds; 90 percent of the tasks are completed in less than 10 seconds, the mean of the distribution. Usually, a user receives his answer with no noticeable delay between the time the request is made with carrier return and the beginning of the typed answer.

The distribution of time for the total user interaction cycle (the time between the initiation of successive tasks) is shown in Figure 13. The mean interaction time is 34 seconds and the median 11 seconds.

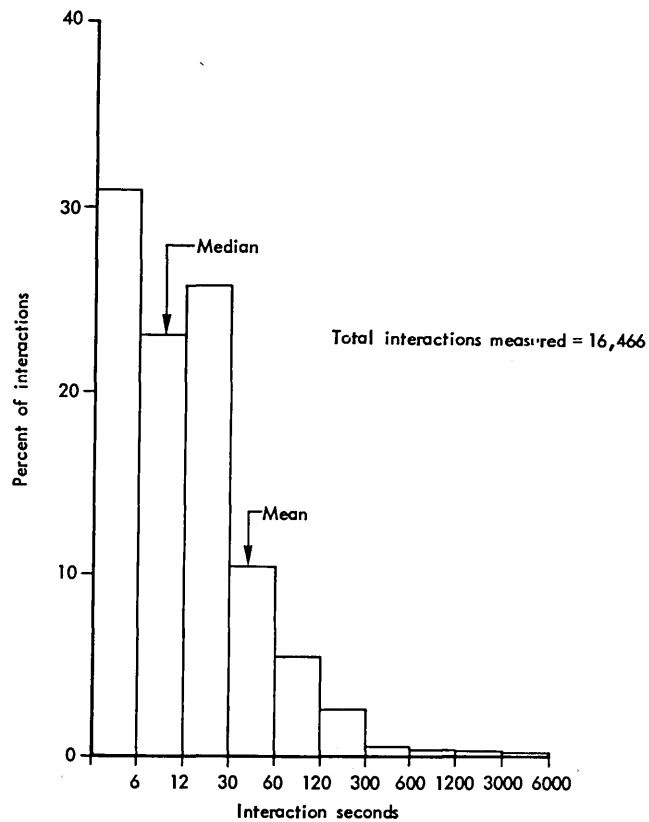


Figure 13—Interaction time

As lines of input and output information flow back and forth between JOSS and the user, the length of these lines, in characters, is recorded. Typical distributions are shown in Figures 14 and 15.

Peaks always appear in the output line-length distribution. The first one, between 0 and 5 characters, is caused by brief, commonly occurring error messages; the peak between 5 and 20 characters is the result of the output for *Demand* statements (request for input from the user); and other peaks, such as the one in the 30 to 35 range, are caused by specific programs producing a large amount of output during the measurement period, or by JOSS unformatted output.

We can now put together an interaction cycle that represents the "average" case. Since the median or "typical" case is so different in character, we also show that breakdown. In fact, in the real system we rarely see the average case because of the form of the distributions, while we often see the typical cases. Average and typical user interactions are shown in Figure 16. It is clear that a well-designed time-sharing system should be able to provide instant response for the typical user interaction even in the face of large

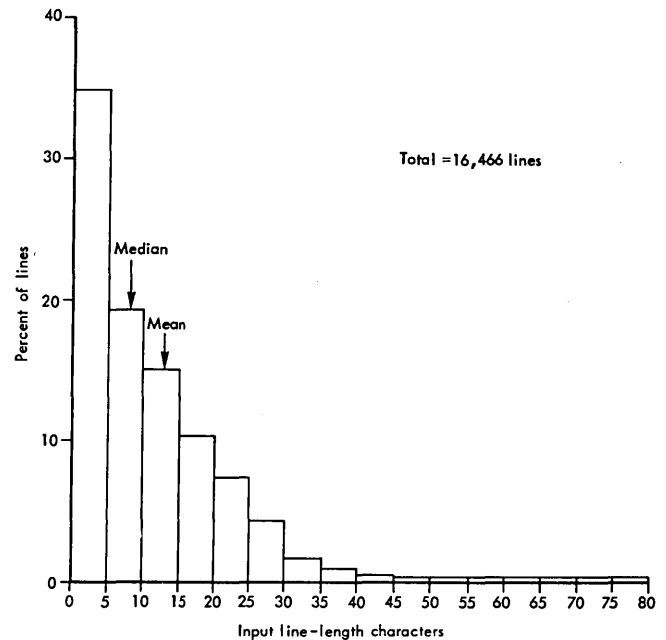


Figure 14—Characters per input line

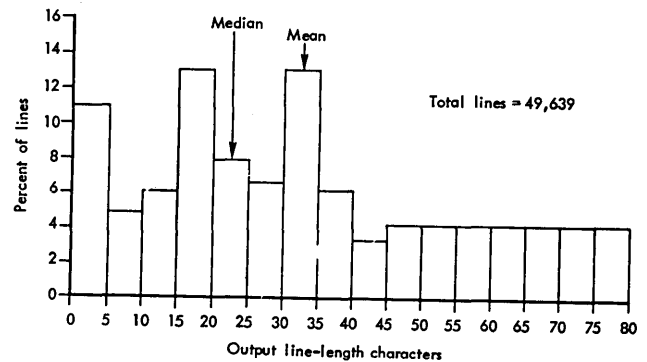


Figure 15—Characters per output line

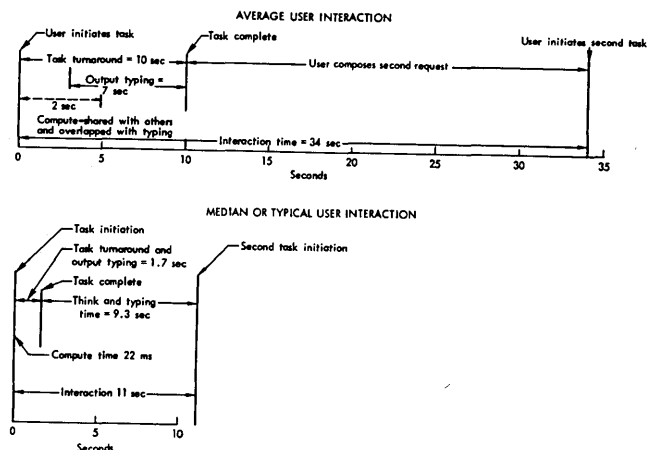


Figure 16—User interactions

amounts of compute time required for other users and any necessary swaps required to bring the user into high-speed memory. The average user, on the other hand, must take a delay of 1 second to receive his 2 seconds of computing.

CONCLUSION

The characteristics of a typical JOSS user session are as follows:

- Time at the console: 46 minutes.
- Compute time: 2.6 minutes.
- The user inputs 82 lines, creating a subtask for each line.
- For each task, 2.9 output lines are produced, the mean compute time is 1.85 seconds, but the median compute time is 22 milliseconds, and 90 percent complete in less than 154 milliseconds.
- Mean elapsed time between input lines is 32 seconds, distributed exponentially, and the output for the task is complete (task turned around) in 9 seconds. The median task turnaround time is 1.9 seconds.
- During the session, 15,000 JOSS statements are executed, and 68,000 arithmetic operations are performed.
- The user recalls 3 items from the files, discards 1 item, and saves 1 item.
- Mean program size is 650 words, but 50 percent are less than 200 words and 10 percent are larger than 2000 words.

The requests that users make of the JOSS system are substantially different from those made on general-purpose, on-line, time-shared systems. In JOSS, there are a relatively large number of requests for short amounts of computing and a relatively small number of requests for a large amount of computing—although the amount of computing is by no means trivial, as can be seen from the number of statements and arithmetic operations performed.

By providing a single easy-to-use language applicable to a wide class of problems and by strict limitations on program size and I/O speed (typewriter only, no tapes), JOSS has created a user environment in which very fast response can be given to typical user requests with simple I/O-based priority scheduling coupled to a simple round-robin scheduling of compute-bound users.

There are several distinctly different types of usage of this system:

1. Desk calculator.
2. Interactive problem solving.
3. Production computing.
4. Output-limited computing.
5. Multiconsole interactive games.

In the future, it may be desirable to report usage statistics by these types.

REFERENCES

- 1 G E BRYAN
JOSS: Introduction to the system implementation
The RAND Corporation Santa Monica Calif P-3486
November 1966 also published by The Digital Equipment Computer Users Society DECUS Proceedings
Fall 1966
 - 2 G E BRYAN
Dynamic characteristics of computer programs
The RAND Corporation Santa Monica Calif p 3661
August 1967
 - 3 A L SCHERR
An analysis of time-shared computer systems
MACTR-18 thesis MIT June 1965 see also "Time sharing measurement" *Datamation* vol 12 No 4
April 1966 pp 22-26
 - 4 R A TOTSCHEK
An empirical investigation into the behavior of the SDC time-sharing system
System Development Corporation Santa Monica Calif
August 1965
 - 5 G E BRYAN
JOSS: User scheduling and resource allocation
The RAND Corporation Santa Monica Calif RM-5216-PR January 1967
RAND Publications on JOSS
- Baker, C. L., *JOSS: Console Design* (RM-5218-PR, February 1967)
- , *JOSS: Introduction to a Helpful Assistant* (RM-5058-PR, July 1966).
- , *JOSS: Rubrics* (P-3560, March 1967).
- , *JOSS: Scenario of a Filmed Report* (RM-4162-PR, June 1964); see also "The JOSS System: Time-Sharing at RAND," *Datamation*, Vol. 10, No. 11 (November 1965), pp. 32-36 (article based on RM-4162-PR).
- Bryan, G. E., *JOSS: Accounting and Performance Measurement* (RM-5217-PR, June 1967).
- , *JOSS: Assembly Listing of the Supervisor* (RM-5437-PR, August 1967).
- , *JOSS: Introduction to the System Implementation* (P-3486, November 1966); also published by The Digital Equipment Computer Users Society, *DECUS Proceedings* (Fall 1966).
- , *JOSS: User Scheduling and Resource Allocation* (RM-5216-PR, January 1967).
- Bryan, G. E., and E. W. Paxson, *The JOSS Notebook* (RM-5367-PR, August 1967).
- Bryan, G. E., and J. W. Smith, *JOSS Language (Apercu and Précis, Pocket Précis, Poster Précis)* (RM-5377-PR, August 1967).
- Gimble, E. P., *JOSS: Problem Solving for Engineers* (RM-5322-PR, May 1967).
- Greenwald, I. D., *JOSS: Arithmetic and Function Evaluation Routines* (RM-5028-PR, September 1966).
- , *JOSS: Console Service Routines (The Distributor)* (RM-5044-PR, September 1966).

- _____, *JOSS: Disc File System* (RM-5257-PR, February 1967).
- Marks, S. L., and G. W. Armerding, *The JOSS Primer* (RM-5220-PR, August 1967).
- Shaw, J. C., *JOSS: Conversations with the Johnniac Open-Shop System* (P-3146, May 1965).
- _____, *JOSS: A Designer's View of an Experimental On-Line Computing System* (P-2922, August 1964); also in *AFIPS Conference Proceedings* (1964 FJCC), Vol. 26 (Spartan Books, Washington, D.C., 1964), pp. 455-464.
- _____, *JOSS: Examples of the Use of an Experimental On-Line Computing Service* (P-3131, April 1965).
- _____, *JOSS: Experience with an Experimental Computing Service for Users at Remote Typewriter Consoles* (P-3149, May 1965).
- Smith, J. W., *JOSS: Central Processing Routines* (RM-5270-PR, August 1967).